

What is claimed is:

1. A system comprising:
a central processor;
a machine-readable device coupled with the central processor, the machine-readable device storing legacy firmware; and
non-volatile memory coupled with the central processor, the non-volatile memory storing legacy-free firmware providing a legacy-free boot path from system power up to operating system loading that is independent of the legacy firmware, wherein the legacy-free firmware includes an optional termination of legacy-free drivers and initiation of a legacy boot using the legacy firmware.
2. The system of claim 1, wherein the non-volatile memory comprises random access non-volatile memory.
3. The system of claim 1, wherein the machine-readable device comprises an option read only memory in a plug-in peripheral component interconnect card.
4. The system of claim 1, wherein the machine-readable device comprises an option read only memory in the non-volatile memory.

5. The system of claim 1, wherein the central processor comprises a central processing unit housed in a single chip.

6. The system of claim 5, further comprising:
a volatile memory; and
a motherboard coupling the volatile memory, the non-volatile memory and the machine-readable device with the central processing unit.

7. A machine-readable material embodying information indicative of instructions for causing a machine to perform operations comprising:

loading a legacy-free driver during an initial boot process using legacy-free firmware;
checking a legacy boot option;
stopping the legacy-free driver; and
loading legacy firmware.

8. The machine-readable material of claim 7, further comprising running the legacy firmware.

9. The machine-readable material of claim 8, wherein the legacy firmware comes from a legacy option read only memory in a plug-in peripheral component interconnect card.

10. The machine-readable material of claim 9, wherein the legacy-free driver comprises an Extensible Firmware Interface driver, and the legacy-free firmware comprises Extensible Firmware Interface firmware.

11. A machine-implemented method for supporting legacy operating system booting in a legacy-free system, the method comprising:

loading a legacy-free driver during an initial boot process using legacy-free firmware;

checking a legacy boot option;

stopping the legacy-free driver; and

loading legacy firmware.

12. The machine-implemented method of claim 11, further comprising running the legacy firmware.

13. The machine-implemented method of claim 12, wherein the legacy firmware comes from a legacy option read only memory in a plug-in peripheral component interconnect card.

14. The machine-implemented method of claim 13, wherein the legacy-free driver comprises an Extensible

Firmware Interface driver, and the legacy-free firmware comprises Extensible Firmware Interface firmware.

15. A data processing system comprising:
means for processing instructions and data; and
means for separating a legacy-free boot path from a legacy boot path such that the legacy boot path comprises an encapsulated boot process.

16. The system of claim 15, wherein the legacy boot path comprises legacy firmware from a legacy option read only memory.

17. The system of claim 16, wherein the legacy option read only memory comprises a part of a plug-in device.

18. The system of claim 16, wherein the legacy option read only memory comprises a part of a legacy platform.